

# A lightweight spatiotemporal graph dilated convolutional network for urban sensor state prediction

Peixiao Wang <sup>a,b</sup>, Hengcai Zhang <sup>a,b,\*</sup>, Shifen Cheng <sup>a,b</sup>, Tong Zhang <sup>c</sup>, Feng Lu <sup>a,b,e</sup>, Sheng Wu <sup>d,e</sup>

<sup>a</sup> State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China

<sup>b</sup> College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China

<sup>c</sup> State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

<sup>d</sup> Key Lab of Spatial Data Mining & Information Sharing of Ministry of Education, The Academy of Digital China (Fujian), Fuzhou University, Fuzhou 350002, China

<sup>e</sup> Fujian Collaborative Innovation Center for Big Data Applications in Governments, Fuzhou 350003, China

## ARTICLE INFO

### Keywords:

Urban computing  
Causal dilated convolution  
Graph dilated convolution  
Spatiotemporal prediction

## ABSTRACT

Spatiotemporal prediction is one attractive research topic in urban computing, which is of great significance to urban planning and management. At present, there are many attempts to predict the spatiotemporal state of systems using various deep learning models. However, most existing models tend to improve prediction accuracy with larger parameter scale and time consumption, but ignoring ease of use in practice. To overcome this question, we propose a lightweight spatiotemporal graph dilated convolutional network called STGDN with satisfactory prediction accuracy and lower model complexity. More specifically, we propose a novel dilated convolution operator and integrate it into traditional causal convolutional networks and graph convolutional networks to greatly improve the efficiency of prediction. The proposed dilated convolution operator can significantly reduce the depth of the model, thereby reducing the parameter scale and improving the computational efficiency of the model. We conducted on multi experiments on three real-world spatiotemporal datasets (traffic dataset, PM<sub>2.5</sub> dataset, and temperature dataset) to prove the effectiveness and advantage of our proposed STGDN. The experimental results show that the proposed STGDN model outperforms or achieves comparable prediction accuracy of the existing nine baselines with higher operational efficiency and fewer model parameters. Codes are available at anonymous private link on <https://doi.org/10.6084/m9.figshare.23935683>.

## 1. Introduction

With the rapid development of the Internet of Things, urban sensor data show explosive growth, such as traffic data, air quality data, and meteorological data (Liu et al., 2023b; Zhang & Farooq, 2023; Zhang et al., 2022a, 2022b). For these urban sensor data with fixed geographical locations, how to construct accurate and reliable spatiotemporal prediction models is a fundamental research topic and one of the key challenges in urban computing (Jin et al., 2023; Liang et al., 2018; Wang et al., 2024).

Existing spatiotemporal prediction models can be roughly divided into knowledge-driven and data-driven spatiotemporal prediction models (McMillan et al., 2023; Wang et al., 2022a; Zhang et al., 2023). Compared with the knowledge-driven spatiotemporal prediction

models, the data-driven spatiotemporal prediction models have become the mainstream prediction methods due to their high prediction accuracy (Fang et al., 2021a; Janowicz et al., 2020). The data-driven models establish a nonlinear function mapping between input and output data by training complex machine learning or deep learning models, thereby significantly improving the prediction accuracy (Xu et al., 2021; Zhang et al., 2022a). However, most existing models focus on improving the prediction accuracy, but ignore the ease of use (Cheng et al., 2020). The improvement of prediction accuracy is often accompanied by an increase in model complexity, which results in the difficulty of model implementation and makes the models computationally inefficient/parameter oversized (Wang et al., 2023). In real-world scenarios, improving prediction accuracy without significantly increasing time consumption and model parameter scale is paramount in urban

\* Corresponding author at: State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China.

E-mail address: [zhanghc@lreis.ac.cn](mailto:zhanghc@lreis.ac.cn) (H. Zhang).

computing (Zheng et al., 2014). The spatiotemporal prediction models are challenging to balance the accuracy of model prediction and ease of use.

The causal dilated convolution network (CDC) provides a potential solution for lightweight spatiotemporal prediction tasks (Bai et al., 2018). Previous studies have shown that the CDC model has the advantages of high prediction accuracy, low time consumption and small parameter scale (Yan et al., 2020). However, the CDC model is mainly used for time series prediction tasks currently rather than spatiotemporal prediction tasks (especially spatiotemporal prediction tasks based on graph structures). Dealing with the spatial dependence in data is a big obstacle for the CDC model. In this paper, we propose a novel spatiotemporal graph dilated convolutional network (STGDN) based on CDC for spatiotemporal prediction tasks. The main contributions of this study are summarized as follows:

- (1) The proposed STGDN is a lightweight spatiotemporal prediction model rather than a time series prediction model. It inherits the advantages of the CDC model and directly serve the spatiotemporal prediction of urban sensor states.
- (2) Similar to the CDC operator, a novel graph dilated convolution operator is designed. The graph dilated convolution operator can effectively and quickly capture spatial dependencies in data without significantly increasing the computation time or parameter scale.
- (3) We used the traffic, PM<sub>2.5</sub> and temperature datasets to evaluate the prediction performance of the STGDN model, including prediction accuracy, computation time and parameter scale. In addition, we opened the code of the STGDN model to ensure the reproducibility of the experimental results.

## 2. Literature review

In this section, we reviewed knowledge-driven and data-driven spatiotemporal prediction models, and analyzed the shortcomings of existing models.

### 2.1. Knowledge-driven spatiotemporal prediction models

Knowledge-driven spatiotemporal prediction models assume that spatiotemporal data obey explicit mathematical laws in spatial or temporal dimension, so as to build specific parametric expression to predict the future spatiotemporal state (Campbell & Thompson, 2008). For example, Kriging interpolation uses the covariance function to obtain an optimal linear unbiased estimate of the unknown spatiotemporal state based on the second-order smoothness of the spatial distribution (Pesquer et al., 2011). Kalman filtering assumes that the variance of the observed data is fixed in the time dimension, and uses linear equations to establish a functional mapping between the observed data and the unknown data (Guo et al., 2014). Autoregressive integrated moving average (ARIMA) assumes that the observed data satisfy temporal smoothness in the time dimension and relies on observations from previous times to infer the future spatiotemporal state (Yozgatligil et al., 2013). In addition, many scholars have proposed hybrid spatiotemporal prediction models based on ARIMA and Kriging models, such as spatiotemporal kriging (ST-Kriging) (Aryaputera et al., 2015) and spatiotemporal ARIMA (ST-ARIMA) (Duan et al., 2016). Although the above knowledge-driven prediction models can be used for spatiotemporal prediction tasks, the prediction accuracy of the models is poor. First, the knowledge-driven spatiotemporal prediction models are based on strict prior assumptions, and the actual spatiotemporal environment is often challenging to meet the premise assumptions of the model. Second, the knowledge-driven spatiotemporal prediction models are parametric models, and it is difficult for parametric models to describe the complex nonlinear relationships in spatiotemporal data (Wang et al., 2020).

### 2.2. Data-driven spatiotemporal prediction models

With the rapid development of artificial intelligence and high-performance computing, data-driven spatiotemporal prediction models have become the mainstream prediction methods (Liu et al., 2023a). Unlike knowledge-driven models, data-driven spatiotemporal prediction models do not require the dataset to obey specific mathematical laws, but use non-parametric operation (such as machine learning or deep learning) to predict future spatiotemporal states (Li et al., 2022; Zhang et al., 2023), such as the spatiotemporal k-nearest neighbor (ST-KNN) (Cheng et al., 2018; Wu et al., 2014), Bayesian temporal matrix factorization model (BTMF) (Chen & Sun, 2022), spatiotemporal residual network model (ST-ResNet) (Zhang et al., 2017), regression updated semi-supervised learning method (RUSSL) (Jiang et al., 2022), and hybrid CNN\_LSTM\_WDTD model (Mengfan et al., 2022). In recent years, given the universality of graph structures, graph convolutional neural networks (GCN) have been gradually applied to spatiotemporal data modeling and further improve the accuracy of prediction tasks (Niepert et al., 2016). Typical graph-based spatiotemporal prediction models include temporal graph convolutional network (T-GCN) (Zhao et al., 2020), bidirectional spatiotemporal graph network (BiSTGN) (Wang et al., 2022b), attention based spatiotemporal graph convolutional neural network (ASTGCN) (Guo et al., 2019), subgraph partitioning and multi-scale GNN (SGMS-GNN) model (Liu et al., 2023b), and dynamic spatiotemporal aware graph neural network (DSTAGNN) (Lan et al., 2022). Although the above models have achieved superior accuracy in spatiotemporal prediction tasks, the high complexity of the models increases the difficulty of implementation and makes the models computationally inefficient/parameter oversized. Most existing spatiotemporal prediction models still struggle to balance the prediction accuracy and ease of use.

Therefore, we propose a novel lightweight spatiotemporal prediction model, i.e., the STGDN model. It aims to improve the prediction accuracy without significantly increasing the computation time and model parameter scale.

## 3. Preliminaries

**Definition 1 (Graph).** As shown in Fig. 1 (a) and (b), the study area can be abstracted as a graph structure  $G = \langle V, E, A \rangle$ , where  $V = \{v_i\}_{i=1}^n$  represents  $n$  sensors in  $G$  ( $n$  graph nodes),  $E$  indicates the relationship between graph nodes. For simplicity, the connection relationships between sensors can be represented by an adjacency matrix  $A \in \mathbb{R}^{n \times n}$ , where  $A_{ij}$  indicates the connection relationship between node  $v_i$  and node  $v_j$ .

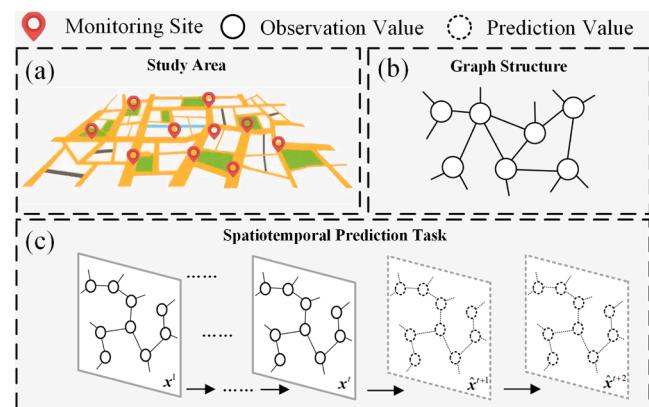


Fig. 1. Related definitions: (a) study area, (b) graph structure, and (c) spatiotemporal prediction task.

**Definition 2 (Spatiotemporal State Matrix).** The spatiotemporal data monitored by all sensors in all time windows can be expressed as a spatiotemporal state matrix  $X \in \mathcal{R}^{n \times T}$ , where  $x_i^t$  indicates the spatiotemporal value (traffic volume and air quality per unit time) monitored by sensor  $v_i$  in the  $t$ th time window,  $x^t = \{x_i^t\}_{i=1}^n \in \mathcal{R}^{n \times 1}$  represents the spatial sequence monitored by all sensors in the  $t$ th time window.

As shown in Fig. 1 (c), our research goal is to establish a functional model  $\mathcal{F}(\cdot)$  that can mine spatiotemporal patterns in the matrix  $X$  based on graph structure  $G$  and accurately predict future spatiotemporal data. Specifically, given a spatiotemporal state matrix  $X$ , the process is shown in Eq. (1).

$$\hat{x}^{t+q} = \mathcal{F}(X'_{t-p+1}, G; W) \quad (1)$$

where,  $\mathcal{F}(\cdot)$  represents the proposed STGDN model;  $X'_{t-p+1} = \{x^t\}_{t=t-p+1}^t \in \mathcal{R}^{n \times p}$  represents the historical spatiotemporal data that the STGDN model needs to input;  $p$  represents the time-dependent step;  $\hat{x}^{t+q}$  represents future spatiotemporal data;  $q$  represents the prediction step,  $q = 1$  indicates single-step prediction, and  $q > 1$  indicates multi-step prediction;  $W$  represents learnable parameters in the STGDN model.

#### 4. Methodology

In this section, we describe the proposed STGDN model for spatiotemporal prediction. As shown in Fig. 2, the proposed STGDN model mainly consists of multiple STGDNCel blocks. Among them, each STGDNCel block contains a causal dilated convolutional module (CDC), and a graph dilated convolutional module (GDC), where the CDC module is used to mine temporal correlations in urban sensor data (discussed in Section 4.1), and the GDC module is used to mine spatial correlation relationships in urban sensor data (discussed in Section 4.2). More specifically, the process of the STGDN model can be described as follows. The original spatiotemporal data is used as the input of the CDC module to obtain a temporary state. Then, the temporary state is used as the input of the GDC module to obtain the output of the STGDNCel block, and the output of the last STGDNCel block is the final prediction result. In addition, to improve the nonlinear fitting ability of the model, we introduce the skip connection (i.e., the  $1 \times 1$  convolution) within the CDC module, and between the original spatiotemporal data and output

of the GDC module.

##### 4.1. Causal dilated convolution module

Existing time correlation mining models can be divided into iterative and non-iterative models. Specifically, iterative models include recurrent neural networks (RNN) and their variants (Chung et al., 2014; Shi et al., 2015; Zhao et al., 2020). Non-iterative models include causal convolutional neural network (Causal CNN) and their variants (Bai et al., 2018; Wang et al., 2019; Yu et al., 2018). The two types of models have advantages and disadvantages. For example, iterative models such as RNN have fewer model parameters but slower computational efficiency (iterative models allow model parameter sharing but can only perform serial operations). In comparison, non-iterative models such as Causal CNN have faster computational efficiency but more model parameters (non-iterative models allow parallel operations but require deeper network layers). To simultaneously reduce the time complexity (improve model computational efficiency) and the space complexity (reduce model parameter scale) of the model, inspired by Zhang et al. (2021), we use a novel lightweight causal dilated convolutional network to mine the time correlation in data, namely the CDC module in STGDN.

The core of the CDC module is the causal dilated convolution operator. Compared with the causal convolution operator, the causal dilated convolution operator can significantly reduce the depth of the model, thereby reducing the parameter scale of the model. Taking the time series  $\{x_i^t\}_{t=t-p+1}^t \in \mathcal{R}^{1 \times 9}$  containing nine timestamps as an example, Fig. 3 further illustrates the difference between causal convolution and causal dilated convolution. If we use convolutional kernel  $\{w_k\}_{k=1}^3 \in \mathcal{R}^{1 \times 3}$  to perform causal convolution operations on time series  $\{x_i^t\}_{t=t-p+1}^t \in \mathcal{R}^{1 \times 9}$ , we need to establish a four-layer causal convolutional neural network. If we perform causal dilation convolution operations with dilation factor  $d$ , we only need to establish a two-layer causal dilated convolutional neural network. Therefore, the causal dilated convolution operator reduces the parameter scale by reducing the depth of the neural network. Taking time series  $\{x_i^t\}_{t=t-p+1}^t \in \mathcal{R}^{1 \times p}$  as an example, the forward propagation process of the causal dilated convolution operator is shown in Eq. (2).

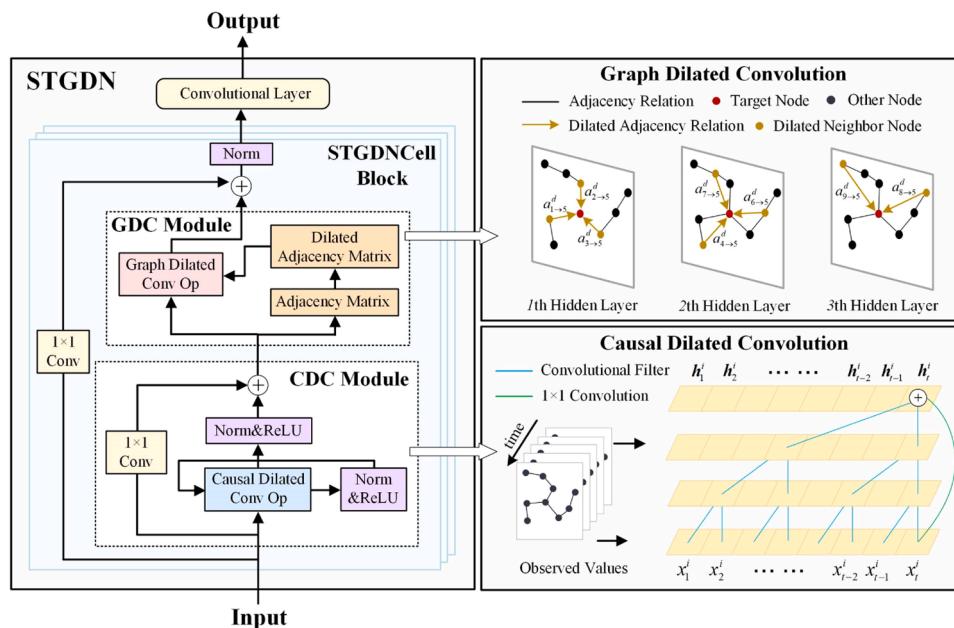
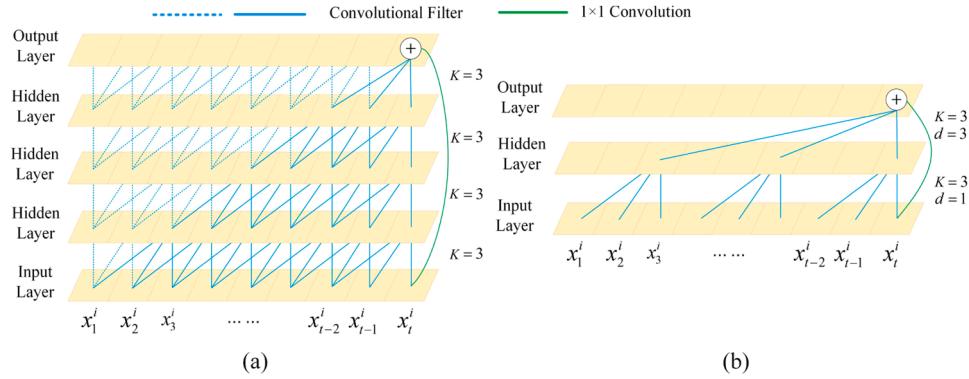


Fig. 2. Workflow of the STGDN model.



**Fig. 3.** Forward propagation of causal convolution and causal dilated convolution: (a) causal convolution, and (b) causal dilated convolution.

$$(F *_d \{x_i^\tau\}_{\tau=t-p+1}^t)_{(x_i^\tau)} = \sum_{k=1}^K w_k x_i^{\tau-(K-k)d} \quad (2)$$

where,  $(F *_d \{x_i^\tau\}_{\tau=t-p+1}^t)_{(x_i^\tau)}$  denotes a causal dilated operation on  $x_i^\tau$  with dilation factor  $d$ ;  $x_i^\tau$  represents the observed value of graph node  $v_i$  in the  $\tau$ th time window;  $\{w_k\}_{k=1}^K$  represents convolution kernel;  $K$  represents the size of the convolutional kernel.

Based on the causal dilated convolution operator, we further define the forward propagation process of the CDC module. In the CDC module, inspired by He et al. (2016), we use skip connection and parametric regularization to improve the prediction accuracy of the model. Taking the time series  $\{x_i^\tau\}_{\tau=t-p+1}^t \in \mathcal{R}^{1 \times p}$  as an example, the forward propagation process of the CDC module is shown in Eq. (3).

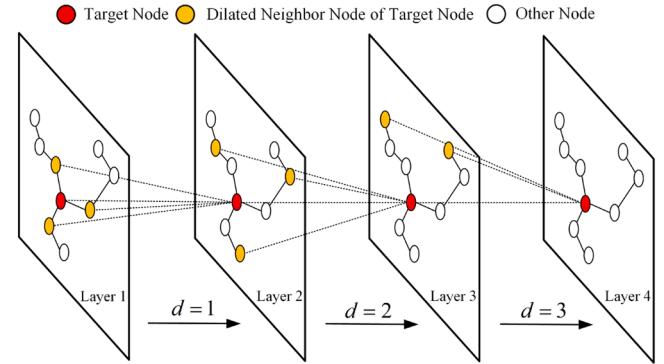
$$CDC\left(\{x_i^\tau\}_{\tau=t-p+1}^t\right) : \begin{cases} \mathbf{H}'_i = F *_d \{x_i^\tau\}_{\tau=t-p+1}^t \\ \mathbf{H}''_i = \text{Relu}(\text{Norm}(\mathbf{H}'_i)) \\ \mathbf{H}_i = \text{Relu}(\text{Norm}(F *_d \mathbf{H}''_i)) + F * \{x_i^\tau\}_{\tau=t-p+1}^t \end{cases} \quad (3)$$

where,  $CDC(\{x_i^\tau\}_{\tau=t-p+1}^t)$  represents the forward propagation of CDC module on input data;  $\mathbf{H}'_i, \mathbf{H}''_i \in \mathcal{R}^{e_h \times p}$  represents the temporary state of the graph node  $v_i$  in the CDC module;  $\mathbf{H}_i$  represents the output state of the graph node  $v_i$  in the CDC module;  $e_h$  represents the number of convolutional kernels;  $F *_d \{x_i^\tau\}_{\tau=t-p+1}^t$  represents causal dilated convolution operator;  $F * \{x_i^\tau\}_{\tau=t-p+1}^t$  represents causal convolution operator, used for residual connections;  $\text{Norm}$  indicates parameter regularization function;  $\text{Relu}$  indicates the activation function.

#### 4.2. Graph dilated convolution module

After the CDC module, a spatiotemporal tensor  $\mathcal{H} = \{\mathbf{H}_i\}_{i=1}^n \in \mathcal{R}^{n \times e_h \times p}$  can be obtained for all graph nodes. To mine the spatial correlation in spatiotemporal tensors, we propose a novel lightweight graph dilated convolutional network, namely the GDC module in STGDN. The core of the GDC module is the graph dilated convolution operator. Similar to the causal dilated convolution operator, the graph dilated convolution operator can also significantly reduce the parameter scale of the model by reducing the depth of the model. As the dilation factor  $d$  increases, it is possible to model long-range spatial dependencies using fewer layers of the neural network.

As shown in Fig. 4, the difficulty of the graph dilated convolution operator lies in efficiently discovering the dilated adjacency relationships of graph nodes. In practice, the dilated neighbors of the target node with an expansion factor  $d$  can be obtained by the matrix  $A^d$ . Specifically, we can find the dilated neighbors of the target node through the position where matrix  $A^d$  is equal to 1. As shown in Fig. 5, through matrix  $A^2$ , we can find that the dilated neighbors of target node  $v_5$  are  $v_4$ ,



**Fig. 4.** Dilated adjacency relationships of the target node: when the dilation factor  $d = 1$ , the dilated adjacent neighbors are the first-order topological neighbors of the target node. when the dilation factor  $d = 2$ , the dilated adjacent neighbors are the second-order topological neighbors of the target node.

$v_6$ , and  $v_7$  with dilation factor  $d = 2$ . Similarly, when the dilation factor is three, the dilated neighbors of target node  $v_5$  are  $v_4$  and  $v_6$ . Based on this property, the method of calculating the dilated adjacency matrix with dilation factor  $d$  can be defined by Eq. (4).

$$A_{ij}^{\text{dilation}=d} = \begin{cases} 1 & i = j \\ 0 & i \neq j \text{ and } A_{ij}^d \neq 1 \\ 1 & i \neq j \text{ and } A_{ij}^d = 1 \end{cases} \quad (4)$$

where,  $A^{\text{dilation}=d}$  represents the dilated adjacency matrix;  $A_{ij}^{\text{dilation}=d}$  represents the dilated adjacency between graph node  $v_i$  and graph node  $v_j$ ;  $d$  represents the dilation factor;  $A^d$  represents the  $d$  power of the first-order topological adjacency matrix.

After obtaining the dilated adjacency matrix, we further defined the graph dilated convolution operator. Inspired by graph attention (Veličković et al., 2018), the graph dilated convolution operator aggregates spatial information through weighting. Specifically, the calculation method for the graph dilated convolution operator is shown in Eq. (5).

$$(F * G_d)_{(v_i)} : \begin{cases} \mathbf{O}_i = \sum_{j \in A_i^{\text{dilation}=d}} \gamma_{ji} \mathbf{H}_i \mathbf{W} \\ \gamma_{ji} = \frac{\exp(\text{Relu}([\mathbf{H}_i \| \mathbf{H}_j] \mathbf{W}_q))}{\sum_{k \in A_i^{\text{dilation}=d}} \exp(\text{Relu}([\mathbf{H}_i \| \mathbf{H}_k] \mathbf{W}_q))} \end{cases} \quad (5)$$

where  $(F * G_d)_{(v_i)}$  represents the graph dilated convolution operation with dilation factor  $d$  for graph node  $v_i$ ;  $\mathbf{H}_i$  is the  $\mathbf{H}_i$  in Eq. (3);  $A_i^{\text{dilation}=d}$

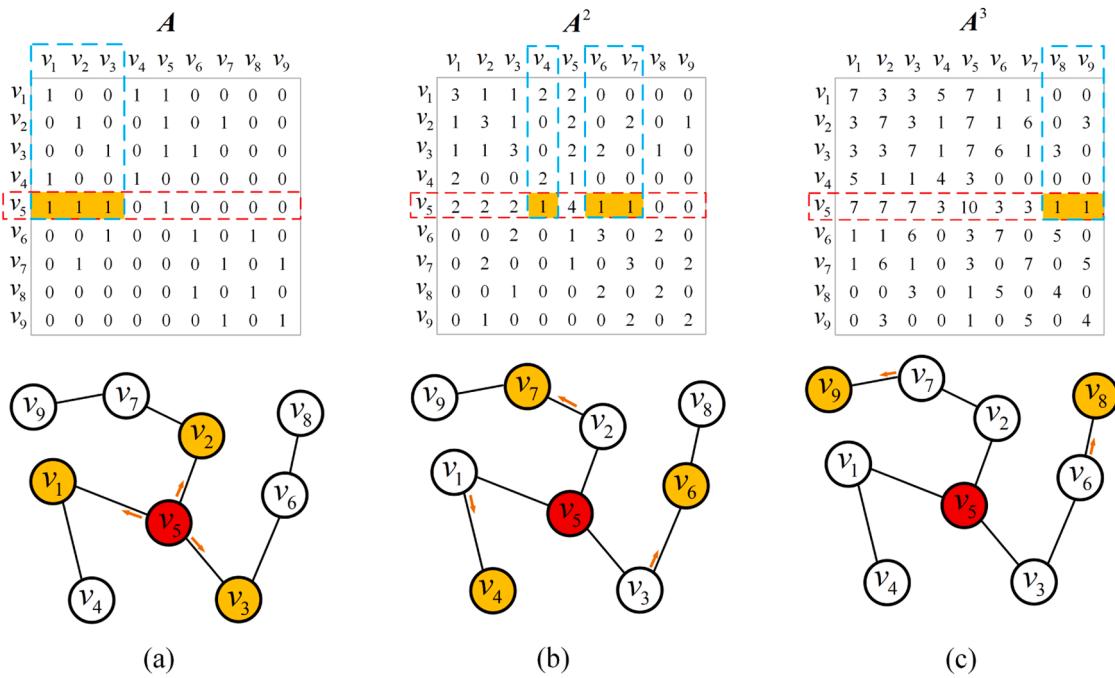


Fig. 5. Calculation of dilated adjacency matrix: (a)  $d=1$ , (b)  $d=2$ , and (c)  $d=3$ .

represents the dilated adjacency matrix with dilation factor  $d$ ;  $W_q$  and  $W_v$  represent the learnable parameter;  $\text{Relu}$  indicates the activation function;  $\exp$  represents the exponential function;  $[\cdot] \cdot [\cdot]$  stands for matrix concatenation function.

Based on the graph dilated convolution operator, we further define the forward propagation of the GDC module. As the GDC module contains only one graph dilated convolution operator, the forward propagation of the graph dilated convolution operator is the forward propagation process of the GDC module, as shown in Eq. (6).

$$\text{GDC}(\{\mathbf{H}_i\}_{i=1}^n) = (F * G_d) \quad (6)$$

where,  $\mathcal{H} = \{\mathbf{H}_i\}_{i=1}^n \in \mathcal{R}^{n \times e_h \times p}$  represents a spatiotemporal tensor, which is the output of the CDC module.

#### 4.3. Optimization of the STGDN

The output of a single STGDNCell will be obtained from the spatiotemporal data  $X_{t-p+1}^t = \{x^{t-p+1}, x^{t-p+2}, \dots, x^t\}$  through the CDC module and the GDC module, and the final prediction result can be obtained from the output of the last STGDNCell. Assume that  $\mathcal{O} = \{\mathbf{O}_i\}_{i=1}^n \in \mathcal{R}^{n \times e_h \times p}$  is the output result of the last STGDNCell, and the final prediction result is shown in Eq. (7).

$$\{\hat{x}^{t+1}, \hat{x}^{t+2}, \dots, \hat{x}^{t+q}\} = F * \mathcal{O} = F * \{\mathbf{O}_i\}_{i=1}^n \quad (7)$$

where,  $\{\hat{x}^{t+1}, \hat{x}^{t+2}, \dots, \hat{x}^{t+q}\}$  represents the final prediction result of the STGDN model;  $q$  represents the prediction step;  $F * \mathcal{O}$  represents the convolution operation on the spatiotemporal tensor  $\mathcal{O} = \{\mathbf{O}_i\}_{i=1}^n \in \mathcal{R}^{n \times e_h \times p}$ ,  $p$  represents the time-dependent step.

The STGDN model predicts future spatiotemporal data  $\{\hat{x}^{t+1}, \hat{x}^{t+2}, \dots, \hat{x}^{t+q}\}$  through historical spatiotemporal data  $X_{t-p+1}^t = \{x^{t-p+1}, x^{t-p+2}, \dots, x^t\}$ . In model optimization, we use the mean square error to optimize the loss between the prediction value  $\{\hat{x}^{t+1}, \hat{x}^{t+2}, \dots, \hat{x}^{t+q}\}$  and the ground truth  $\{x^{t+1}, x^{t+2}, \dots, x^{t+q}\}$ . The loss function of the STGDN model is shown in Eq. (8).

$$\mathcal{L}(\mathbf{W}) = \min_{\mathbf{W}} \left( \sum_{j=1}^q \| \hat{\mathbf{x}}^{t+j} - \mathbf{x}^{t+j} \|_2^2 \right) \quad (8)$$

where,  $\mathbf{x}^{t+j} \in \mathcal{R}^{n \times 1}$  represents the ground truth of all node within the  $(t+j)$ th time window;  $\hat{\mathbf{x}}^{t+j} \in \mathcal{R}^{n \times 1}$  represents the prediction values of all node within the  $(t+j)$ th time window.

#### 4.4. Algorithm and training

In Sections 4.1–4.3, we have discussed the forward propagation and backpropagation of the STGDN model in detail. In this section, we further introduce the training process of the STGDN model. The basic principle of the STGDN model is to establish a supervised learning method, using the spatiotemporal patterns contained in the historical spatiotemporal data to predict future spatiotemporal data. To train the STGDN model, we divide the spatiotemporal data into training samples and test samples, where the training samples are used to train the parameters of the STGDN model, and the test samples are used to test the prediction performance of the STGDN model. Algorithm 1 shows the training process of the STGDN model. First, we build training instances of the model  $\mathcal{M}$  based on the spatiotemporal state matrix (lines 1–3). Second, based on the training instances, we obtain the prediction results of the STGDN model (lines 6–10). Finally, the STGDN model is obtained by optimizing the error loss until the model converges (line 11).

### 5. Experimental results and analysis

#### 5.1. Data preparation

##### 5.1.1. Data sources

Three spatiotemporal datasets are used to evaluate the prediction performance of the STGDN model, namely, traffic, PM<sub>2.5</sub> and temperature dataset. Table 1 shows the statistical characteristics of three datasets.

The traffic dataset comes from 71 monitoring cameras in Wuhan, China, and records the traffic volume of a single camera at a specific time window (the time window size is 5 min). Fig. 6 (a) shows the spatial

**Algorithm 1**

Training process of STGDN.

---

**Require:** Spatiotemporal state matrix:  $X = \{x_t\}_{t=1}^T$   
 Time dependent step:  $p$   
 Prediction step:  $q$   
 Number of STGDNCell:  $n_b$

**Ensure:** STGDN model:  $\mathcal{M}$   
 //construct training instances of STGDN

- 1:  $\mathcal{D} \leftarrow \emptyset$
- 2: **for** next  $t \in [p, 2, \dots, T - q]$  **do**
- 3: put a training instance  $(\{x_{t-p+1}, \dots, x_t\}, \{x_{t+1}, \dots, x_{t+q}\})$  into  $\mathcal{D}$
- //train STGDN model
- 4: initialize the parameters  $W$  of STGDN
- 5: **repeat**
- 6: randomly select a training instance  $\mathcal{D}_b$  from  $\mathcal{D}$
- 7: **for** next  $i \in [1, 2, \dots, n_b]$  **do**
- 8: if  $i = 1$ :  $data_{in} = \mathcal{D}_b$  else:  $data_{in} = \emptyset$
- 9: obtain spatiotemporal tensor  $\mathcal{O}$  by Eqs. (5), (6), and  $data_{in}$
- 10: obtain  $(\hat{x}^{t+1}, \hat{x}^{t+2}, \dots, \hat{x}^{t+q})$  by Eq. (7)
- 11: find  $W$  by minimizing the Eq. (8)
- 12: **until**  $\mathcal{M}$  converges
- 13: output the learned models  $\mathcal{M}$

---

**Table 1**

Description of the datasets.

Dataset	Traffic volume	PM2.5	Temperature
Location	Wuhan	Beijing	Beijing
Size of time window	5 min	60 min	60 min
Number of spatial objects	71	36	45
Number of temporal objects	8064	2208	2184
Time span	2021/3/1–2021/3/28	2014/10/1–2014/12/31	2023/4/1–2023/6/30

distribution of monitoring cameras. Each traffic data contains the unique identification of the monitoring camera, the coordinates of the monitoring camera, the monitoring time window, and the traffic volume within the time window.

The PM<sub>2.5</sub> dataset comes from 36 air quality monitoring stations in Beijing, China (the time window size is 60 min) (Zheng et al., 2015). Fig. 6(b) shows the spatial distribution of air quality monitoring stations. Each record contains the unique identification of the monitoring station, the coordinates of the monitoring station, the monitoring time window, and the PM<sub>2.5</sub> concentration within the time window.

The temperature dataset comes from the Copernicus climate database (Hersbach et al., 2018), and records the air temperature at 2 m above the surface of inland waters (the time window size is 60 min). As shown in Fig. 6(c), we select 45 grids in Beijing for the experiment, and the grid resolution is 0.25° × 0.25°. Each temperature data contains the

unique identification of the grid, the center coordinates of the grid, the monitoring time window, and the average temperature in the time window.

**5.1.2. Data preprocessing**

To support this work, we preprocessed three spatiotemporal datasets as follows:

- (1) There are natural missing values in the collected spatiotemporal data. Considering the impact of natural missing values on subsequent modeling, we used the BTTF model to estimate the natural missing values in spatiotemporal data (Chen & Sun, 2022).
- (2) We constructed first-order topological adjacency matrices for three spatiotemporal datasets. In this study, the first-order topological adjacency matrix is constructed by a similarity matrix, and we use the ten most similar spatial objects as the first-order topological neighbors of the target spatial object.
- (3) The manually processed data were divided into training and test instances. According to the 20–80 criterion, the training instances account for 80 %, and the test instances account for 20 %.

**5.2. Evaluation metrics**

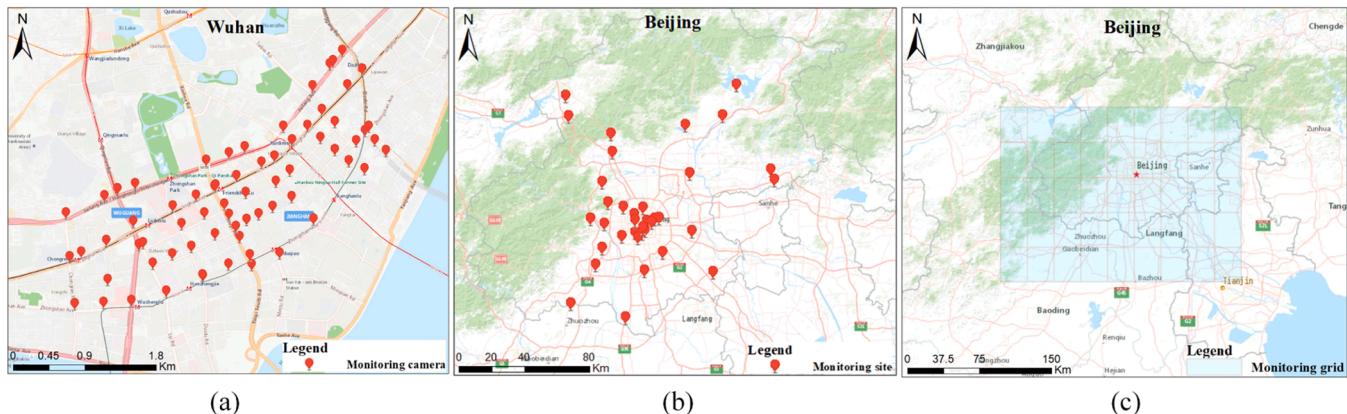
In the spatiotemporal prediction task, a critical problem is how to evaluate the performance of the prediction model. In this study, the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) are used as quantitative indicators to verify the prediction accuracy of the proposed STGDN model. The calculation methods for MAE, RMSE, and MAPE are shown in Eqs. (9)–(11).

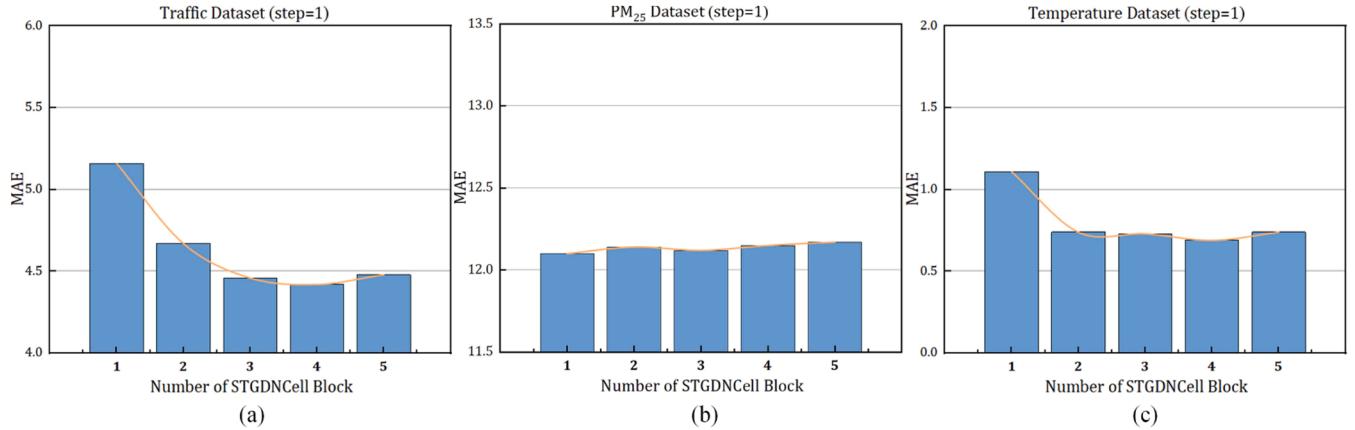
$$MAE = \frac{1}{n * q} \sum_{i=1}^n \sum_{j=1}^q |x_i^{t+j} - \hat{x}_i^{t+j}| \quad (9)$$

$$RMSE = \sqrt{\frac{1}{n * q} \sum_{i=1}^n \sum_{j=1}^q (x_i^{t+j} - \hat{x}_i^{t+j})^2} \quad (10)$$

$$MAPE = \frac{100}{n * q} \sum_{i=1}^n \sum_{j=1}^q \left| \frac{x_i^{t+j} - \hat{x}_i^{t+j}}{x_i^{t+j}} \right| \quad (11)$$

where,  $x_i^{t+j}$  represents the ground truth of graph node  $v_i$  within the  $(t+j)$ th time window;  $\hat{x}_i^{t+j}$  represents the prediction value of graph node  $v_i$  within the  $(t+j)$ th time window;  $n$  represents the total number of graph nodes in the study area;  $q$  represents the prediction step.

**Fig. 6.** Study area: (a) monitoring cameras in traffic flow data, (b) monitoring sites in PM2.5 data, and (c) experimental grids in temperature data.



**Fig. 7.** Parameter tuning of STGCNCell Block: (a) traffic dataset, (b) PM<sub>2.5</sub> dataset, and (c) temperature dataset.

### 5.3. Hyper-parameter selection

In this subsection, we describe the experimental environment (hardware and software environment) and hyper-parameter setting information.

In this study, the spatiotemporal data is processed on a PC (Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, memory: 32.0GB). Moreover, we built our model based on PyTorch and Python3.7 on a Graphics Processing Unit (GPU) platform with 24GB of GPU memory.

The hyper-parameters of the STGDN model mainly include the time-dependent step  $p$  and number of STGDNCell  $n_b$ . In this study, the control variable method is used to obtain the optimal combination of hyper-parameters. More specifically, Fig. 7 shows the calibration process for the number of STGDNcells in three datasets. In the traffic and temperature datasets, the indicator MAE shows a trend of first decreasing and then stabilizing with the increase of  $n_b$ . In the PM<sub>2.5</sub> dataset, as  $n_b$  increases, the trend of MAE changes is not significant. The result indicates that the prediction results in the traffic and the temperature datasets are influenced by long-term dependence, whereas the prediction results in the pm<sub>2.5</sub> data are mainly influenced by short-term dependence. At the same time, through the change curve of MAE, we can determine the value of  $n_b$ , namely  $n_b=3$  in the traffic dataset,  $n_b=2$  in the PM<sub>2.5</sub> dataset, and  $n_b=2$  in the temperature dataset.

### 5.4. Comparison with baselines

As the prediction performance of knowledge-driven models is often lower than that of data-driven models, we mainly compare STGDN model with data-driven models. There are nine baseline methods which can be roughly divided into two categories. The first category is machine learning models, including ST-KNN model (Zheng & Su, 2014) and BTMF model (Chen & Sun, 2022). The second category is deep learning models, including T-GCN model (Zhao et al., 2020), BiSTGN model (Wang et al., 2022b), STGODE model (Fang et al., 2021b), STA-ODE model, GDGCN model (Xu et al., 2023), ASTGCN model (Guo et al., 2019), and DSTAGNN model (Lan et al., 2022). For the ASTGCN model, we use only recent component to perform prediction tasks.

#### 5.4.1. Comparison results of prediction accuracy

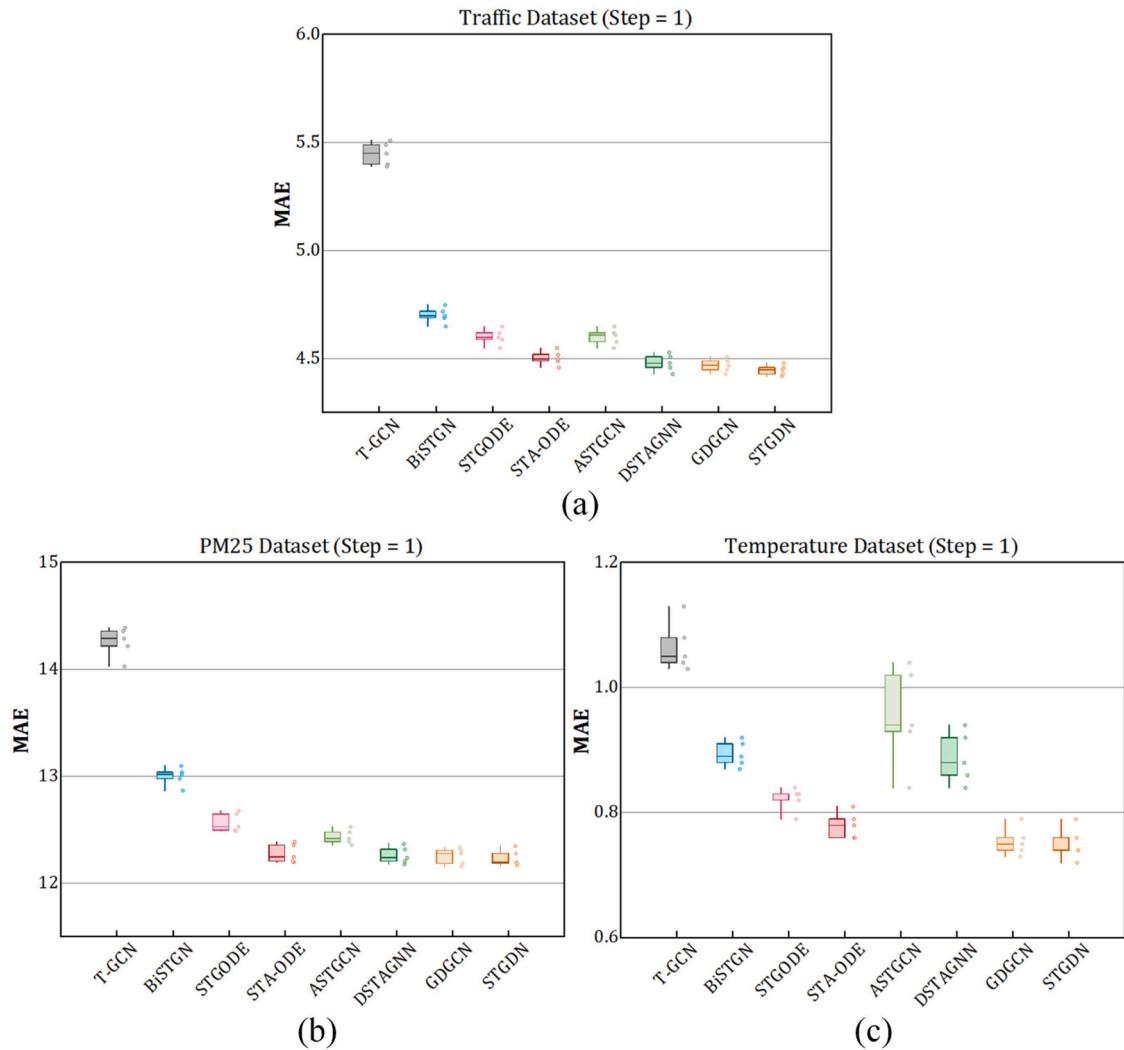
The comparison results of the prediction accuracy between the STGDN model and the baselines are shown in Table 2. The results show that on the three datasets, the prediction accuracy of the second type model is higher than that of the first type model, i.e., the prediction performance of deep learning models is higher than that of machine learning. Specifically, on the traffic and PM<sub>2.5</sub> datasets, the prediction accuracy of the STGDN model is higher than that of the ST-KNN, BTMF, T-GCN, BiSTGN, STGODE, ASTGCN, and DSTAGNN models, and is close to the

**Table 2**

Comparison results (in MAE/RMSE/MAPE) of prediction accuracy between STGDN and baselines.

Model	Traffic Volume		PM 2.5		Temperature	
	1-step	3-steps	1-step	3-steps	1-step	3-steps
ST-KNN	5.44/	5.81/	23.80/	28.92/	1.45/	1.71/
	8.69/	9.37/	41.11/	48.82/	1.94/	2.61/
	24.25	25.64	62.66	83.60	5.52	6.49
BTMF	6.41/	6.70/	18.59/	24.37/	1.12/	1.55/
	10.48/	10.98/	32.23/	40.83/	1.49/	1.97/
	35.41	36.3	50.04	74.53	4.29	5.83
T-GCN	5.49/	5.57/	14.29/	23.03/	1.08/	1.40/
	11.39/	11.9/	30.44/	45.92/	1.43/	1.80/
	23.29	25.89	32.09	55.26	3.89	5.09
BiSTGN	4.72/	4.96/	13.04/	21.65/	0.87/	1.27/
	8.32/	8.36/	24.99/	39.75/	1.17/	1.66/
	20.41	23.91	31.56	52.69	3.18	4.68
STGODE	4.65/	4.88/	12.53/	20.40/	0.82/	1.19/
	7.45/	7.82/	24.28/	38.27/	1.13/	1.56/
	20.82	23.13	29.13	52.83	3.01	4.39
STA-ODE	4.50/	4.82/	12.20/	20.09/	0.78/	1.09/
	7.09/	7.74/	23.78/	37.63/	1.07/	1.45/
	19.97	21.75	28.99	52.35	2.88	4.10
ASTGCN	4.61/	4.85/	12.53/	20.33/	1.02/	1.51/
	7.45/	7.75/	24.94/	38.15/	1.39/	1.95/
	20.43	22.27	30.77	52.26	3.66	5.47
DSTAGNN	4.48/	4.81/	12.24/	19.92/	0.94/	1.45/
	7.05/	7.71/	23.32/	37.23/	1.31/	1.87/
	19.32	21.86	28.70	51.82	3.37	5.26
GDGCN	4.47/	4.80/	12.19/	19.82/	0.74/	1.07/
	7.03/	7.69/	23.43/	37.23/	1.04/	1.41/
	19.61	21.03	29.70	50.73	2.72	4.01
STGDN	4.46/	4.80/	12.17/	19.84/	0.74/	1.07/
	7.03/	7.67/	23.10/	37.42/	1.03/	1.42/
	19.09	21.17	28.52	51.32	2.72	4.03

prediction accuracy of the STA-ODE, DSTAGNN, and GDGCN models. On the temperature dataset, the prediction accuracy of the STGDN model is higher than that of the ST-KNN, BTMF, T-GCN, BiSTGN, STGODE, ASTGCN, and DSTAGNN models, and is close to the prediction accuracy of the STA-ODE and GDGCN models. In addition, we further analyzed the prediction accuracy of STGDN and benchmark experiments under different random seeds, as shown in Fig. 8. Among them, the dots represent the prediction accuracy of the model under different random seeds, and the box line represents the center position and spread range of the dot. The results indicate that the proposed STGDN model has relatively stable prediction accuracy under different random seeds and can obtain the same conclusions as before. The above results indicate that the proposed STGDN model has achieved or outperformed the baseline prediction accuracy without considering the computational efficiency and model parameter scale.



**Fig. 8.** Prediction accuracy under different random seeds: (a) traffic dataset, (b) PM2.5 dataset, and (c) temperature dataset.

#### 5.4.2. Comparison results of model complexity

In this subsection, we further analyzed the model complexity of the STGDN model. Considering that the STGDN model is a deep learning model, we only compared the STGDN model with seven deep learning methods.

Table 3 shows the comparison results of the running time between the STGDN model and the baselines. Among them, the time required for forward and backward propagation (FBP) is mainly used to reflect the offline training speed of the STGDN model, and the time required for forward propagation (FP) is mainly used to reflect the online prediction speed of the STGDN model. The results show that compared to STGODE, STA-ODE, ASTGCN, DSTAGNN, and GDGCN models, the proposed STGDN model has significant advantages in forward propagation time, indicating that the STGDN model is expected to be applied to online prediction applications with high real-time requirements. In addition, compared to early T-GCN and BiSTGN models, the proposed STGDN model did not significantly increase the computational time of the model, further proving that the proposed STGDN model has significant advantages in computational efficiency. In addition to the model runtime, Fig. 9 shows the advantages of the STGDN model in terms of model parameter scale. The results indicate that the proposed STGDN model has superior prediction accuracy and fewer model parameters, proving that the proposed STGDN model has lower spatial complexity than baselines.

In general, the proposed STGDN model outperformed or achieved

comparable prediction accuracy of baselines with faster time efficiency and fewer model parameters.

#### 5.5. Qualitative analysis of prediction results

In this subsection, we qualitatively analyzed the spatial and temporal distribution of prediction errors using line charts and maps. Fig. 10 shows the temporal distribution characteristics of prediction errors in three datasets. The results show that the prediction error of the three-step prediction is slightly higher than that of the single-step prediction (the gray area at the bottom of the line chart). The cumulative effect of errors often causes the large errors in multi-step prediction. In addition, the area with large prediction error is mainly concentrated where the curve fluctuates violently (blue area). The reason why the error in the blue region is large is that the sudden change of the observed values in a short time tends to increase the difficulty of the prediction model. For example, in the traffic dataset, the traffic volume during rush hour is more difficult to predict than during off-peak hours, which aligns with our common sense.

Fig. 11 further shows the spatial distribution characteristics of prediction errors in three datasets. Similar to the prediction error in the temporal dimension, the areas with large prediction errors are mainly concentrated in locations with large data fluctuations in the spatial dimension. For example, areas with large prediction errors in traffic datasets are mainly concentrated on main roads. The reason is that the

**Table 3**

Comparison results (milliseconds, ms) of running time between STGDN and baselines (batch size is 64).

Model	Traffic Volume		PM 2.5		Temperature		
	1-step	3-steps	1-step	3-steps	1-step	3-steps	
T-GCN	FP (ms)	36.4 ±2.39	45.3 ±2.13	18.9 ±2.36	23.2 ±2.34	22.6 ±3.41	27.6 ±3.61
	FBP (ms)	63.5 ±2.77	80.1 ±3.42	32.9 ±2.77	42.1 ±2.63	39.1 ±3.68	47.9 ±4.01
BiSTGN	FP (ms)	54.8 ±3.04	68.2 ±3.52	27.8 ±2.47	36.4 ±2.85	32.9 ±3.10	42.5 ±2.17
	FBP (ms)	70.5 ±3.84	87.2 ±4.13	40.5 ±3.11	52.4 ±3.02	46.1 ±3.64	57.6 ±3.27
STGODE	FP (ms)	138.4 ±5.38	178.3 ±8.21	71.3 ±3.14	90.1 ±5.14	93.4 ±6.43	116.2 ±7.18
	FBP (ms)	207.6 ±10.8	251.2 ±11.8	95.6 ±7.23	124.6 ±6.74	134.6 ±8.19	165.4 ±9.24
STA-ODE	FP (ms)	310.8 ±11.4	388.4 ±12.4	40.3 ±2.84	51.4 ±3.03	95.2 ±4.32	123.5 ±6.18
	FBP (ms)	360.4 ±13.2	453.6 ±14.1	61.8 ±3.43	79.2 ±3.62	113.4 ±6.16	143.4 ±8.13
ASTGCN	FP (ms)	226.9 ±5.46	235.2 ±5.89	22.6 ±1.02	22.8 ±1.07	75.2 ±8.98	77.8 ±9.24
	FBP (ms)	253.6 ±6.67	262.2 ±6.57	31.8 ±1.03	32.1 ±1.12	93.7 ±9.42	95.6 ±10.1
DSTAGNN	FP (ms)	329.8 ±12.4	334.2 ±11.7	39.9 ±2.64	41.3 ±2.28	128.9 ±10.4	128.9 ±10.4
	FBP (ms)	379.8 ±14.3	386.4 ±12.6	59.6 ±3.83	61.3 ±3.42	169.6 ±11.8	175.3 ±11.6
GDGCN	FP (ms)	152.5 ±6.18	159.3 ±8.42	83.7 ±5.34	88.2 ±6.03	105.7 ±9.18	118.7 ±8.24
	FBP (ms)	475.7 ±12.8	493.2 ±13.2	254.3 ±8.23	262.1 ±8.65	315.1 ±12.4	338.2 ±14.3
STGDN	FP (ms)	77.43 ±1.67	79.89 ±2.01	7.91 ±0.69	8.12 ±0.87	25.2 ±2.12	26.9 ±2.19
	FBP (ms)	126.7 ±3.91	129.7 ±3.89	16.2 ±1.00	16.4 ±1.04	48.9 ±4.55	49.4 ±4.60

\* The format of the numbers is mean ± standard deviation, i.e., mean ± std.

traffic volume on the main road fluctuates greatly, making it difficult for the model to predict accurately. In the PM<sub>2.5</sub> and temperature datasets, the areas with large prediction errors are mainly concentrated in the main urban areas of Beijing, China. The main reason is that the main urban area is the main area for human activities. Human activities may cause significant fluctuations in PM<sub>2.5</sub> levels and temperatures, making it difficult to predict accurately. In addition, we further calculated the correlation coefficient between prediction error and error standard deviation to quantify the relationship between data fluctuations and prediction difficulty. The results indicate that there is a significant positive correlation between error standard deviation and prediction accuracy on the three datasets, proving that large observation fluctuations may lead to low prediction accuracy. The above results are also consistent with the findings of Wang et al. (2023). When fluctuations are large, the

spatiotemporal patterns in the data are also more complex. Overly complex spatiotemporal patterns may not be captured by the proposed model. In addition, larger fluctuations may also introduce additional observational errors that may worsen the model's predictions.

Overall, the proposed STGDN model has superior prediction accuracy and accurately predicts the trend of the three datasets in both the time dimension and the spatial dimension.

### 5.6. Effects of different modules on prediction results

In this section, we further analyzed the impact of different modules on the prediction results, as shown in Table 4. Among them, CDC represents the CDC module in the STGDN model, and GDC represents the GDC module in the STGDN model. The results show that the prediction accuracy of the CDC module is better than that of the GDC module, meaning that the prediction results are more affected by the temporal correlation than the spatial correlation. In addition, the prediction accuracy of the STGDN model is better than that of the CDC module, indicating that introducing spatiotemporal correlation is beneficial for improving the prediction accuracy of the model. At the same time, the results also demonstrate the rationality of introducing spatiotemporal correlation.

## 6. Discussion

Spatiotemporal prediction is one attractive research topic in urban computing, which is significant to urban planning and intelligent transportation. However, most existing spatiotemporal prediction models are challenging to balance prediction accuracy and model ease of use. Specifically, although most of the existing spatiotemporal prediction models have high prediction accuracy, the high complexity of the existing models not only increases the difficulty of the model implementation, but also makes the models computationally inefficient and parameter oversized. Therefore, we propose a novel lightweight spatiotemporal prediction model (i.e., the STGDN model) to address the above challenges.

Compared to existing spatiotemporal prediction models, the proposed STGDN model has significant advantages. For machine learning models such as the ST-KNN model and the BTMF model, the proposed STGDN model significantly improves the prediction accuracy of spatiotemporal prediction models. For deep learning models such as the T-GCN model, the BiSTGN model, the STGODE model, the STA-ODE model, the GDGCN model, the ASTGCN model, and the DSTAGNN model, the proposed STGDN model not only achieves superior prediction accuracy, but also simultaneously reduces the model's parameter scale and improves the model's computational efficiency. In addition, we provide a simple implementation of the STGDN model to help users efficiently accomplish spatiotemporal prediction tasks. In conclusion, we can believe that the proposed STGDN model is an advanced

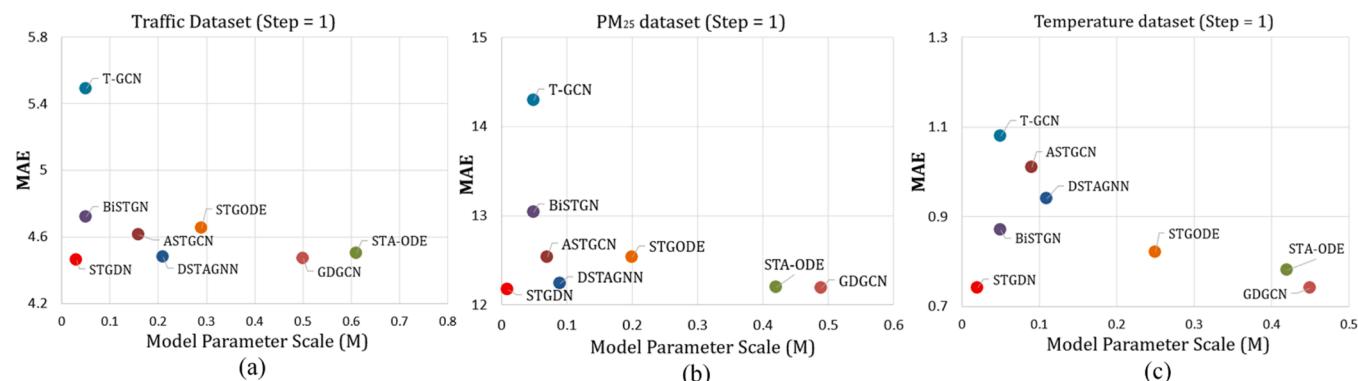
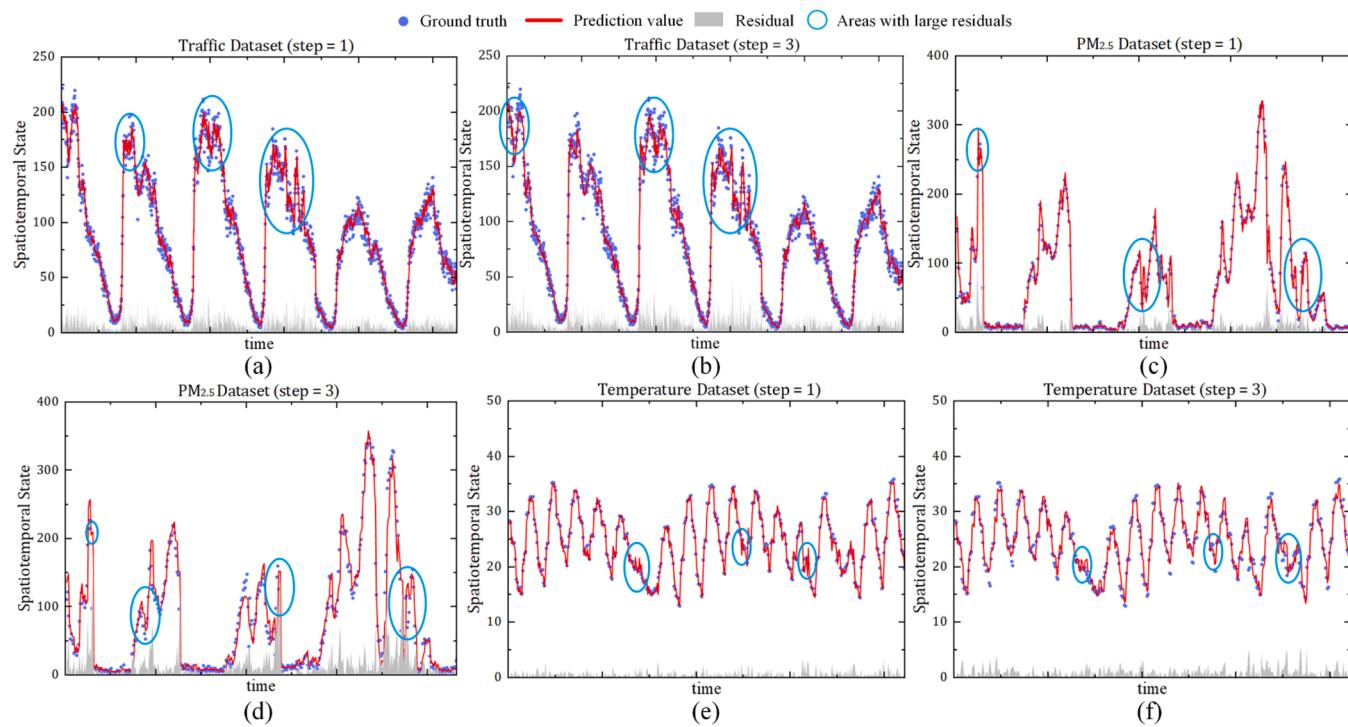
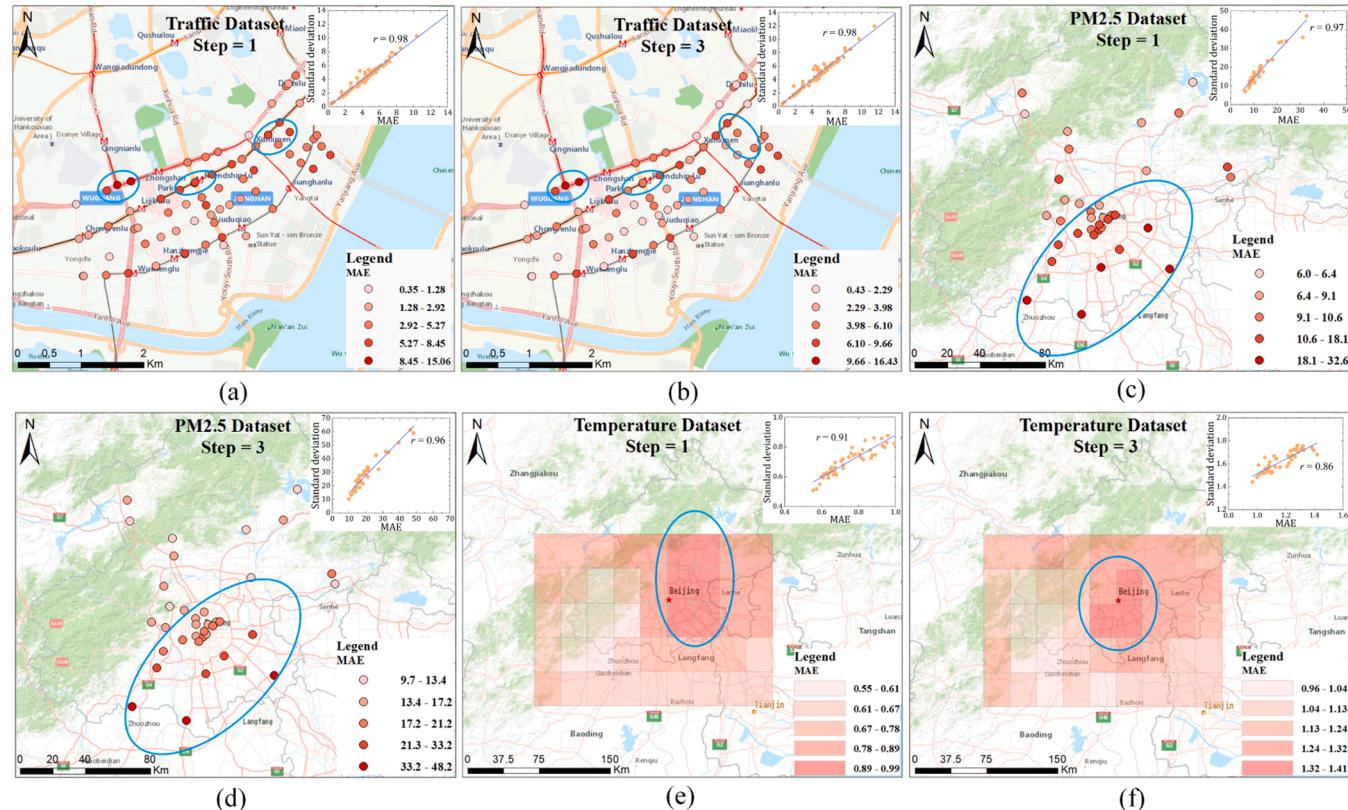


Fig. 9. Model parameter scale vs MAE: (a) traffic dataset, (b) PM2.5 dataset, and (c) temperature dataset.



**Fig. 10.** Prediction error of temporal dimension: (a) one-step prediction on traffic dataset, (b) three-step prediction on traffic dataset, (c) one-step prediction on PM<sub>2.5</sub> dataset, (d) three-step prediction on PM<sub>2.5</sub> dataset, (e) one-step prediction on temperature dataset, and (f) three-step prediction on temperature dataset.



**Fig. 11.** Prediction error of spatial dimension: (a) one-step prediction on traffic dataset, (b) three-step prediction on traffic dataset, (c) one-step prediction on PM<sub>2.5</sub> dataset, (d) three-step prediction on PM<sub>2.5</sub> dataset, (e) one-step prediction on temperature dataset, and (f) three-step prediction on temperature dataset.

**Table 4**

Comparison results (in MAE/RMSE/MAPE) of prediction accuracy between STGDN and baselines.

Model	Traffic Volume		PM 2.5		Temperature	
	1-step	3-steps	1-step	3-steps	1-step	3-steps
CDC	4.77/	4.92/	12.69/	20.30/	0.81/	1.17/
	7.45/	8.08/	23.82/	38.01/	1.23/	1.53/
	21.54	23.23	34.46	57.35	3.42	4.30
GDC	5.01/	5.19/	13.23/	21.05/	0.88/	1.24/
	7.90/	9.68/	24.61/	38.73/	1.28/	1.62/
	23.55	25.89	35.57	60.17	3.66	4.57
STGDN	4.46/	4.80/	12.17/	19.84/	0.74/	1.07/
	7.03/	7.67/	23.10/	37.42/	1.03/	1.42/
	19.59	21.17	28.52	51.32	2.72	4.03

spatiotemporal prediction model.

The limitations of this study are as follows: We only empirically tested the prediction performance of the STGDN model on small graph structures. For example, the traffic dataset has only 71 graph nodes, the PM2.5 dataset has only 36 graph nodes, and the temperature dataset has only 45 graph nodes. However, in actual scenarios, the scale of the graph structure is often large. For example, an urban road network may contain thousands of graph nodes (thousands of road segments). In future work, the prediction performance (prediction accuracy, computational efficiency, and model parameter scale) of the STGDN model will be further validated in large-scale graph structures.

## 7. Conclusion

At present, most existing spatiotemporal prediction models are challenging to balance prediction accuracy and model ease of use. we propose a novel lightweight spatiotemporal prediction model, i.e., the STGDN model. In the experimental section, three real spatiotemporal datasets (traffic dataset, PM2.5 dataset, and temperature dataset) are used to verify the prediction performance of the STGDN model. First, the control variable method was used to obtain the optimal combination of parameters for the STGDN model. Second, we conducted a comparison with nine existing data-driven baselines, including STKNN, BTMF, T-GCN, BiSTGN, STGODE, STA-ODE, GDGCN, ASTGCN, and DSTAGNN models. The experimental results showed that the proposed STGDN model outperformed or achieved comparable prediction accuracy of the existing nine baselines with faster time efficiency and fewer model parameters. Finally, the influence of different components in STGDN on prediction accuracy was tested, proving that the proposed method is suitable for spatiotemporal prediction.

## Funding

This project was supported by National Key Research and Development Program of China [Grant No. 2022YFB3904102], China National Postdoctoral Support Program for Innovative Talents [Grant No. BX20230360], China Postdoctoral Science Foundation [Grant No. 2023M743454], National Natural Science Foundation of China [Grant Nos. 42101423, 42371470, and 42371469], Special Research Assistant Program of Chinese Academy of Sciences, Innovation Project of LREIS [Grant No. 08R8A092YA].

## Declaration of Competing Interest

The authors declare no conflicts of interest.

## Data availability

The data and codes supporting the main findings of this study are available at <https://doi.org/10.6084/m9.figshare.23935683>.

## Acknowledgments

We thank the anonymous referees for their helpful and insightful comments and suggestions.

## References

- Aryaputera, A. W., Yang, D., Zhao, L., & Walsh, W. M. (2015). Very short-term irradiance forecasting at unobserved locations using spatio-temporal kriging. *Solar Energy*, 122, 1266–1278. <https://doi.org/10.1016/j.solener.2015.10.023>
- Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271 [Cs]. <http://arxiv.org/abs/1803.01271>.
- Campbell, J. Y., & Thompson, S. B. (2008). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4), 1509–1531. <https://doi.org/10.1093/rfs/hhm055>
- Chen, X., & Sun, L. (2022). Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 4659–4673. <https://doi.org/10.1109/TPAMI.2021.3066551>
- Cheng, S., Lu, F., Peng, P., & Wu, S. (2018). Short-term traffic forecasting: An adaptive ST-KNN model that considers spatial heterogeneity. *Computers, Environment and Urban Systems*, 71, 186–198. <https://doi.org/10.1016/j.compenvurbsys.2018.05.009>
- Cheng, S., Peng, P., & Lu, F. (2020). A lightweight ensemble spatiotemporal interpolation model for geospatial data. *International Journal of Geographical Information Science*, 34(9), 1849–1872. <https://doi.org/10.1080/13658816.2020.1725016>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555 [Cs]. doi:[10.48550/arXiv.1412.3555](https://doi.org/10.48550/arXiv.1412.3555).
- Duan, P., Mao, G., Zhang, C., & Wang, S. (2016). STARIMA-based traffic prediction with time-varying lags. In *Proceedings of the 2016 IEEE 19th international conference on intelligent transportation systems (ITSC)* (pp. 1610–1615). <https://doi.org/10.1109/ITSC.2016.7795773>
- Fang, Z., Long, Q., Song, G., & Xie, K. (2021a). Spatial-temporal graph ODE networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 364–373). <https://doi.org/10.1145/3447548.3447430>
- Fang, Z., Long, Q., Song, G., & Xie, K. (2021b). Spatial-temporal graph ODE networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 364–373). <https://doi.org/10.1145/3447548.3447430>
- Guo, J., Huang, W., & Williams, B. M. (2014). Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, 43, 50–64. <https://doi.org/10.1016/j.trc.2014.02.006>
- Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In , 33. *Proceedings of the AAAI conference on artificial intelligence* (pp. 922–929). <https://doi.org/10.1609/aaai.v33i01.3301922>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Hersbach, H., Bell, B., Berrißford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., & Thépaut, J.-N. (2018). (2018): ERA5 hourly data on single levels from 1979 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). Copernicus Climate Change Service (C3S) Climate Data Store (CDS). <https://doi.org/10.24381/cds.adbb2d47>
- Janowicz, K., Gao, S., McKenzie, G., Hu, Y., & Bhaduri, B. (2020). GeoAI: Spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. *International Journal of Geographical Information Science*, 34(4), 625–636. <https://doi.org/10.1080/13658816.2019.1684500>
- Jiang, F., Ma, J., & Li, Z. (2022). Pedestrian volume prediction with high spatiotemporal granularity in urban areas by the enhanced learning model. *Sustainable Cities and Society*, 79, Article 103653. <https://doi.org/10.1016/j.scs.2021.103653>
- Jin, G., Liang, Y., Fang, Y., Huang, J., Zhang, J., & Zheng, Y. (2023). Spatio-temporal graph neural networks for predictive learning in urban computing: A survey (arXiv: 2303.14483). arXiv. <https://doi.org/10.48550/arXiv.2303.14483>.
- Lan, S., Ma, Y., Huang, W., Wang, W., Yang, H., & Li, P. (2022). DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *Proceedings of the 39th international conference on machine learning* (pp. 11906–11917). <https://proceedings.mlr.press/v162/lan22a.html>.
- Li, Y., Peng, T., Hua, L., Ji, C., Ma, H., Nazir, M. S., & Zhang, C. (2022). Research and application of an evolutionary deep learning model based on improved grey wolf optimization algorithm and DBN-ELM for AQI prediction. *Sustainable Cities and Society*, 87, Article 104209. <https://doi.org/10.1016/j.scs.2022.104209>
- Liang, Y., Ke, S., Zhang, J., Yi, X., & Zheng, Y. (2018). *GeoMAN: multi-level attention networks for geo-sensor data time series prediction* (pp. 3428–3434). AAAI Press.
- Liu, J., Zhang, T., Gao, C., & Wang, P. (2023a). Forecasting earthquake magnitude and epicenter by incorporating spatiotemporal priors into deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 61, 1–13. <https://doi.org/10.1109/TGRS.2023.3281784>

- Liu, Z., Zhang, S., Shao, X., & Wu, Z. (2023b). Accurate and efficient urban wind prediction at city-scale with memory-scalable graph neural network. *Sustainable Cities and Society*, 99, Article 104935. <https://doi.org/10.1016/j.scs.2023.104935>
- McMillan, L., Fayaz, J., & Varga, L. (2023). Flow forecasting for leakage burst prediction in water distribution systems using long short-term memory neural networks and Kalman filtering. *Sustainable Cities and Society*, 99, Article 104934. <https://doi.org/10.1016/j.scs.2023.104934>
- Mengfan, T., Siwei, L., Ge, S., jie, Y., Lechao, D., hao, L., & Senlin, H. (2022). Including the feature of appropriate adjacent sites improves the PM2.5 concentration prediction with long short-term memory neural network model. *Sustainable Cities and Society*, 76, Article 103427. <https://doi.org/10.1016/j.scs.2021.103427>
- Niepert, M., Ahmed, M., & Kutzkov, K. (2016). Learning convolutional neural networks for graphs. In *Proceedings of the international conference on machine learning* (pp. 2014–2023). <http://proceedings.mlr.press/v48/niepert16.html>.
- Pesquer, L., Cortés, A., & Pons, X. (2011). Parallel ordinary kriging interpolation incorporating automatic variogram fitting. *Computers & Geosciences*, 37(4), 464–473. <https://doi.org/10.1016/j.cageo.2010.10.010>
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., & Woo, W. (2015). Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. In , 1. *Proceedings of the 28th international conference on neural information processing systems* (pp. 802–810).
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks (arXiv:1710.10903). arXiv. <https://doi.org/10.4855/0/arXiv.1710.10903>.
- Wang, J.-H., Lin, G.-F., Chang, M.-J., Huang, I.-H., & Chen, Y.-R. (2019). Real-time water-level forecasting using dilated causal convolutional neural networks. *Water Resources Management*, 33(11), 3759–3780. <https://doi.org/10.1007/s11269-019-02342-4>
- Wang, P., Hu, T., Gao, F., Wu, R., Guo, W., & Zhu, X. (2022a). A hybrid data-driven framework for spatiotemporal traffic flow data imputation. *IEEE Internet of Things Journal*, 9(17), 16343–16352. <https://doi.org/10.1109/JIOT.2022.3151238>
- Wang, P., Zhang, T., Zhang, H., Cheng, S., & Wang, W. (2024). Adding attention to the neural ordinary differential equation for spatio-temporal prediction. *International Journal of Geographical Information Science*, 38(1), 1–26. <https://doi.org/10.1080/13658816.2023.2275160>
- Wang, P., Zhang, T., Zheng, Y., & Hu, T. (2022b). A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation. *International Journal of Geographical Information Science*, 36(6), 1231–1257. <https://doi.org/10.1080/13658816.2022.2032081>
- Wang, P., Zhang, Y., Hu, T., & Zhang, T. (2023). Urban traffic flow prediction: A dynamic temporal graph network considering missing values. *International Journal of Geographical Information Science*, 37(4), 885–912. <https://doi.org/10.1080/13658816.2022.2146120>
- Wang, S., Cao, J., & Yu, P. (2020). Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 1. <https://doi.org/10.1109/TKDE.2020.3025580>. –1.
- Wu, S., Yang, Z., Zhu, X., & Yu, B. (2014). Improved k-nn for short-term traffic forecasting using temporal and spatial information. *Journal of Transportation Engineering*, 140(7), Article 04014026. [https://doi.org/10.1061/\(ASCE\)TE.1943-5436.0000672](https://doi.org/10.1061/(ASCE)TE.1943-5436.0000672)
- Xu, L., Chen, N., Chen, Z., Zhang, C., & Yu, H. (2021). Spatiotemporal forecasting in earth system science: Methods, uncertainties, predictability and future directions. *Earth-Science Reviews*, 222, Article 103828. <https://doi.org/10.1016/j.earscirev.2021.103828>
- Xu, Y., Han, L., Zhu, T., Sun, L., Du, B., & Lv, W. (2023). Generic dynamic graph convolutional network for traffic flow forecasting. *Information Fusion*, , Article 101946. <https://doi.org/10.1016/j.inffus.2023.101946>
- Yan, J., Mu, L., Wang, L., Ranjan, R., & Zomaya, A. Y. (2020). Temporal convolutional networks for the advance prediction of ENSO. *Scientific Reports*, 10(1), 8055. <https://doi.org/10.1038/s41598-020-65070-5>
- Yozgatligil, C., Aslan, S., Iyigun, C., & Batmaz, I. (2013). Comparison of missing value imputation methods in time series: The case of Turkish meteorological data. *Theoretical and Applied Climatology*, 112(1), 143–167. <https://doi.org/10.1007/s00704-012-0723-x>
- Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence* (pp. 3634–3640). <https://doi.org/10.24963/ijcai.2018/505>
- Zhang, B., Cheng, S., Zhao, Y., & Lu, F. (2023). Inferring intercity freeway truck volume from the perspective of the potential destination city attractiveness. *Sustainable Cities and Society*, 98, Article 104834. <https://doi.org/10.1016/j.scs.2023.104834>
- Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI*.
- Zhang, L., Na, J., Zhu, J., Shi, Z., Zou, C., & Yang, L. (2021). Spatiotemporal causal convolutional network for forecasting hourly PM2.5 concentrations in Beijing, China. *Computers & Geosciences*, 155, Article 104869. <https://doi.org/10.1016/j.cageo.2021.104869>
- Zhang, S. R., & Farooq, B. (2023). Interpretable and actionable vehicular greenhouse gas emission prediction at road link-level. *Sustainable Cities and Society*, 92, Article 104493. <https://doi.org/10.1016/j.scs.2023.104493>
- Zhang, T., Liu, J., & Wang, J. (2022a). Rainstorm prediction via a deep spatio-temporal-attributed affinity network. *Geocarto International*, 0(0), 1–19. <https://doi.org/10.1080/10106049.2022.2076914>
- Zhang, Y., Zheng, X., Helbich, M., Chen, N., & Chen, Z. (2022b). City2vec: Urban knowledge discovery based on population mobile network. *Sustainable Cities and Society*, 85, Article 104000. <https://doi.org/10.1016/j.scs.2022.104000>
- Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., & Li, H. (2020). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9), 3848–3858. <https://doi.org/10.1109/TITS.2019.2935152>
- Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3), 1–55. <https://doi.org/10.1145/2629592>
- Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., & Li, T. (2015). Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th SIGKDD conference on knowledge discovery and data mining*. <https://www.microsoft.com/en-us/research/publication/forecasting-fine-grained-air-quality-based-on-big-data/>.
- Zheng, Z., & Su, D. (2014). Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm. *Transportation Research Part C: Emerging Technologies*, 43, 143–157. <https://doi.org/10.1016/j.trc.2014.02.009>